

Evaluation of product-platform decisions based on total supply chain costs

Maud Van den Broeke ^{a,1}, Robert Boute ^{a,b,2}, Behzad Samii ^{a,3}

^a *Technology & Operations Management Area, Vlerick Business School, Belgium*

^b *Research Center for Operations Management, University KU Leuven, Belgium*

1 Corresponding author: Email: maud.vandenbroeke@vlerick.com, Tel.: +32 497 07 91 67

2 Email: robert.boute@kuleuven.be, Tel.: +32 16 24 88 96

3 Email: behzad.samii@vlerick.com, Tel.: +32 92 10 98 50

Evaluation of product-platform decisions based on total supply chain costs

Over the past decades, several companies have introduced product platforms in the design of their products in order to produce a large product variety in a cost-efficient way. However, for some companies, the introduction of platforms ended up being more costly than expected, leading them to reconsider their platform decisions. In this paper we develop a model to support companies in determining (1) how many platforms to develop, (2) which platforms to develop, and (3) which products to derive from which platforms. The model takes into account the impact of these product-platform decisions on a company's relevant supply chain activities and costs. The model shows how the optimal product-platform decisions depend on the trade-off between the costs of platforms versus the costs of customising these platforms to final product variants. We propose a simulated annealing algorithm to solve large problem instances within reasonable time. The practical validity of our model is shown through its application in a global technology company specialised in the development and production of medical screens.

Keywords: product-platform decisions, total supply chain costs, multiple platforms, customisable platforms

1 Introduction

In today's hypercompetitive times, several industries have discovered product platforms as a way to offer a large product variety to their customers without increasing costs and time-to-market (Simpson, et al., 2014). Initially, platforms were primarily used in the automotive, aircraft and electronics industry, but today they have gained importance and proven their use in many other applications. Depending on the business context, platforms can be defined in different ways, ranging from a narrow physical product approach to a combination of common components, processes, knowledge, people and relationships (Robertson & Ulrich, 1998). We follow the approach of Meyer and Lehnerd (1997),

who define a product platform as “a set of subsystems and interfaces that form a common structure from which a stream of derivative products can be efficiently developed and produced.”

Some companies aim at developing one *single* platform from which all product variants can be derived. However, similar to de Weck et al (2003) and Chen and Wang (2008) we believe that the use of *multiple* platforms from which a set of final products is derived, is more likely to offer more cost-efficient solutions. The majority of existing platform literature considers platforms to be either modular or scalable. *Modular* platforms consist of functional modules that can be added, substituted and/or removed to make unique products, whereas *scalable* platforms use scaling variables to stretch or shrink the platform to satisfy a variety of customer needs (Du, et al., 2014). In this paper, however, we consider platforms to be a monolithic (non-modular and non-scalable) part of the final product, from which distinctive product variants can be derived by adding extra components to customise and tailor the platforms to the individual products’ requirements (Agrawal, et al., 2013). We refer to these platforms as ‘*customisable platforms*’. The motivation to analyse customisable platforms (instead of modular or scalable ones) is twofold. Firstly, literature has paid little attention to customisable platforms compared to modular and scalable platforms. Secondly, scalable platforms are technically infeasible for several business cases, including the one described in this paper. Moreover, modular platforms have failed due to their negative impact on the technical product performance (Holttä, et al., 2005).

Existing literature on platforms is rich, ranging from an engineering perspective with the goal to determine the technical configuration of components, modules, platforms and products, to a more business-oriented perspective focusing on the evaluation of platforms in terms of costs and product quality (Jiao, et al., 2007). We contribute to this literature by developing a quantitative model that is capable to support companies to make the following product-platform decisions:

- (1) How many platforms should be developed?
- (2) Which platforms should be developed?
- (3) Which final products should be derived from which platforms?

Rather than evaluating these product-platform decisions from an engineering perspective, we quantify their impact on all relevant costs throughout the supply chain. The contribution of our model lies in the identification and the quantification of how these product-platform decisions impact the major supply chain activities (and costs) within the company. The cost families in our model relate to development, purchasing, inventory management and transformation during production.

- (1) The *development costs* include the cost of designing the platforms and customising them to final products, and the time needed for prototyping, certification and testing.
- (2) The *purchasing costs* reflect the cost of manufacturing, including equipment, tooling, changeovers and facilities. When external suppliers manufacture the platforms – which is a common practice in ever-increasing outsourced manufacturing – the scale economies in platform manufacturing (due to e.g. more commonality and larger buying/production volumes) are reflected by (all-unit) quantity discounts in the purchasing price. The costs of packaging, transportation and distribution of platforms and components purchased from suppliers are reflected in the *ordering costs* of the model.
- (3) The *inventory related costs* cover the cost of storage, warehousing and insurance of cycle and safety inventory, and the cost of inventory shortages.
- (4) Finally, we include the costs to derive and customise final products from the platforms during production, which are denoted as the *transformation costs* (this activity is customarily performed in-house).

The general conviction in literature is that introducing platforms – through the principle of commonality (i.e. products sharing common platforms) – leads to economies of scale and scope in product development, procurement, manufacturing, material and inventory costs (Zhang & Huang, 2010; Collier, 1981). An important finding of our research is that when we include all relevant supply chain costs within a company, introducing platforms does not always lead to a reduction in the total supply chain costs. This can be explained by the underlying trade-off when introducing platforms, namely the trade-off between the costs of platforms versus the costs of customisation (i.e. the costs of

deriving products from those platforms). This trade-off is driven by the number of platforms developed and by the decision whether these platforms are ‘under- or overdesigned’ in relationship to the products derived from it (referring to a platform having a ‘performance level’ lower or higher than the one of the product derived from it (Krishnan & Gupta, 2001)). Using less platforms may reduce platform costs thanks to commonality and economies of scale. Nevertheless, these benefits might be offset by the high capital investments of overdesigned platforms, or in case of underdesigned platforms by the high number of (possibly more expensive) components needed during customisation. These are the implicit trade-offs our model takes into account.

We have validated the applicability of our integrated supply chain cost model to evaluate (and facilitate) product-platform decisions through a real business case in a global technology firm. This company offers medical screens (considered as the final product variants) that can be derived from a set of different types of outsourced manufactured printed circuit boards (considered as the product platforms) by adding extra components during in-house customisation and assembly (such as e.g. connectors, wires and mechanical components).

The remainder of this paper is organized as follows. In section 2 we provide a literature review. In section 3 we present our decision support model and describe the supply chain costs taken into account for the evaluation of the product-platform decisions, and in section 4 we discuss its underlying cost trade-off. Section 5 includes a numerical case study and the solution method to apply our model for practical (large-scale) instances. Section 6 concludes.

2 Literature review

The richness of product platform literature has been illustrated in Jiao et al (2007) and Simpson et al (2014). The reason for its research popularity is that product platforms impact both the company’s costs and product quality, as well as flexibility and time-to-market, in the end impacting the overall company’s profitability (Muffatto, 1999).

Extensive studies are devoted to the optimal design, configuration and architecture of the platforms, its components and modules, and the final products. In particular, Fellini et al (2006) and D’Souza and Simpson (2003) focus on the trade-off between economic efficiency of platforms (i.e.

the cost reductions due to commonality) versus platform effectiveness (i.e. technical product performance) to find the optimal product platform design. Agard and Bassetto (2013) jointly consider cost and product performance (i.e. quality) to select which modules to use in a set of products. De Weck et al (2003) determine the optimal number of product platforms on the basis of the trade-off between commonality and product performance, and the relationship of platforms to different target market segments. Another stream of literature focusses on evaluating the costs associated with a platform strategy. Thyssen et al (2006) and Farrell and Simpson (2010) apply the technique of Activity Based Costing (ABC) to model the impact of modularity and commonality on different activities, which are in turn linked to different costs. Seepersad et al (2000) quantify the costs of labour, material and inventory in terms of scalable platform design variables (such as tube length and tube type) and find that the desirable number of platforms increases as products become more sparsely distributed across the market segment.

While the majority of platform research examines scalable or modular platforms, we consider *customisable* platforms. This resembles the platform research by Ben-Arieh et al (2009), who consider product platforms as the starting point after which components are added or removed to change the features of products. Whereas their research question is similar to ours (e.g. what are the optimal number of platforms and which products should be assigned to which platform), they approach the issue from a technical perspective (i.e. they determine the component set of each platform) and they only examine a limited set of costs (i.e. the cost of components, assembly and adding or removing components). They find that using multiple platforms for the production of a family of products is always cheaper than using a single or no platform. We show that this is not necessarily the case when more supply chain costs are taken into account.

Our model contributes to this stream of literature as it studies a distinct set of product-platform decisions by quantifying their impact on the costs involved in different supply chain activities.

Over the last couple of years the interrelationship between product platform and supply chain design has gained more attention (Zhang, et al., 2010; Yadav, et al., 2011; Khalaf, et al., 2011; Baud-

Lavigne, et al., 2012). Unlike these papers, we do not make supply chain design decisions (e.g. which suppliers to use or where to locate production plants), but these papers inspired us to determine the relevant cost factors in product-platform decisions. Khalav et al (2011) quantify the cost of assembly, storage and transportation to decide in which facilities modules should be produced. Baud-Lavigne et al (2012) consider the interplay between product and component standardisation and supply chain design by modelling the production and transportation costs to determine which products and how many products to manufacture in which production centre.

Our decision support model is closest to Zhang et al (2010) and Zhang and Huang (2010), who quantify the supply chain costs of design and product development, ordering, purchasing (including quantity discounts) and holding inventories. However, our paper differs in the decisions we evaluate (e.g. which products are to be derived from which platforms), and in the way the supply chain costs are expressed (e.g. we additionally consider safety stocks and inventory shortages due to forecast inaccuracies in customer demand). Also, whereas Zhang and Huang (2010) examine modular platforms, we look at customisable platforms. This entails, among other things, that we express supply chain costs on a platform and customisation level. This classification is inspired by Gonzalez-Zugasti et al (2001), who make the distinction between two sets of costs: those for devising platforms (*platform level*) and those for creating variants of those platforms (*customisation level*). This classification uncovers additional insights in the cost behaviour under different product-platform decisions.

3 Problem formulation and supply chain costs evaluation

3.1 Notations

Appendix 1 gives an overview of the notations used in our model (the grey shaded boxes indicate the inputs needed from the firm, whereas the white boxes are decision variables or variables derived from the inputs; observe, given the extended scope of our supply chain cost model, these inputs are to be obtained from different departments within the company, such as sales, production, procurement or R&D). Vectors and matrices are expressed by capital letters, and their elements by the corresponding

small letter. The position of an element in the vector or matrix is expressed by subscripts, whereas superscripts are used for extra clarification: ‘*pl*’ refers to the platforms, ‘*de*’ refers to deriving the final products from the platforms by adding components and ‘*co*’ refers to the components itself.

3.2 Conceptualisation of the model

Our model considers n final products that are offered to the customer and can be derived from m possible customisable platforms by adding c extra components during customisation (see Figure 1). Note that the maximum number of platforms will never exceed the number of products offered to the market ($m \leq n$). Depending on the product-platform combination, different amounts and types of components are required during customisation (for instance, less or cheaper components may be required during customisation in the case of overdesigned platforms or when many platforms are developed). The total number of unique components is c^{max} .

We express a product-platform scenario – which is a set of possible product-platform combinations – by the $m \times n$ matrix S , for which $s_{ij} = 1$ if product j is derived from platform i and zero otherwise. From S we derive which platforms should be developed, expressed by the vector E , for which $e_i = 1$ if platform i is developed and zero otherwise:

$$\forall i; \text{if } \sum_{j=1}^n s_{ij} > 0 \text{ then } e_i = 1; \text{ otherwise } e_i = 0.$$

We assume that a final product can only be derived from one platform at a time and that all products are offered to the market. The use of substitution in platforms may be more cost efficient, but its impact on costs (such as inventory costs) is hard to evaluate (Deflem & Van Nieuwenhuyse, 2013). We take into account the technical feasibility of each product-platform combination by the $m \times n$ matrix T , where $t_{ij} = 1$ if product j can be derived from platform i and is zero otherwise. The number

of product-platform scenarios thus equals $\prod_{j=1}^n \left(\sum_{i=1}^m t_{ij} \right)$.

In matrix S , platforms and products are ranked from low to higher-end (we assume that platforms can be single-criterion-ranked according to the platform development time). This helps to

evaluate whether a platform i is under- or overdesigned for product j (respectively having a ‘performance level’ lower or higher than the product derived from it): when $s_{ij} = 1$ for $i > j$, the platform is overdesigned for product j , it is underdesigned if $i < j$, and $i = j$ refers to a “dedicated” platform i for product j (no over- or underdesign). To evaluate whether or not the introduction of platforms is beneficial, we compare scenarios of platform commonality with the base case scenario of no platform commonality. The base case scenario corresponds to the scenario where all products are derived from their own dedicated platform (i.e. $s_{ij} = 1$ for all $i = j$).

(Please insert Figure 1 about here)

We evaluate each product-platform scenario S by the Net Present Value (NPV) of all relevant supply chain costs related to the platforms and to the customisation to final products (as the final product’s pricing is value-based and not cost-based, it is sufficient to focus on cost minimisation). The initial development costs linked to the development of platforms is denoted by DC^{pl} , and the development of final products from these platforms by DC^{de} . The costs of purchasing and ordering platforms is denoted by PC^{pl} and components needed during customisation PC^{co} . The inventory related costs of the platforms IC^{pl} and components IC^{co} include the cost of cycle and safety inventory, and the cost of inventory shortages. Finally, the transformation costs to customise a platform into a final product during production is represented by TC^{de} (see Figure 2). Whereas the development costs are usually incurred only once (at the beginning), the other costs are yearly recurring over the expected lifetime of platforms (*life*), and are to be discounted using the company’s Weighted Average Cost of Capital (WACC).

This results in the following objective function to determine the optimal product-platform scenario S :

$$\min_S \left[DC^{pl}(S) + PC^{pl}(S) + IC^{pl}(S) + DC^{de}(S) + PC^{co}(S) + IC^{co}(S) + TC^{de}(S) \right] \quad (1)$$

subject to $\forall j; \sum_{i=1}^m s_{ij} = 1$

$$\sum_{i=1}^m \sum_{j=1}^n s_{ij} = n$$

$$s_{ij} \leq t_{ij}$$

(Please insert Figure 2 about here)

3.3 *Supply chain costs impacted by a product-platform scenario*

In this section we elaborate in more detail on the various supply chain costs defined in (1), and how the product-platform decisions impact them.

3.3.1 *Development costs*

The development costs constitute a considerable part of the total costs related to product-platform decisions. The development costs originate from the time (in man-hours) R&D engineers spend on designing, prototyping, certification and testing platforms and products, and their wage cost (w^{en}). The platforms and products are still in a conceptual development stage and are not yet tangible, which they will become subsequently in the production stage. The development costs are only incurred once, meaning that once a platform or product exists no extra development costs will be charged.

We distinguish between the platform development cost DC^{pl} and the customisation development cost DC^{de} . The platform development cost DC^{pl} is driven by the decision about which platforms are to be developed (E) and the time needed to develop each platform (DT^{pl}). When multiple platforms are developed, there may be some economies of scope, which originate from the fact that engineers learn from other platforms, speeding up the platform development. The magnitude of the reduction in development time (DT^{re}) depends on which platforms are jointly developed: the more similarity between platforms, the higher the potential reduction in development time; and the more platforms are jointly developed, the more likely the company will benefit from these economies of scope. The development cost to derive products from platforms, referred to as customisation development cost DC^{de} , is driven by the decision from which platforms the products are derived (S)

and the design time to derive a certain product from a certain platform (DT^{de}). This results in the following expressions for the platform and customisation development costs:

$$DC^{pl}(S) = w^{en} \left[\sum_{i=1}^m (dt_i^{pl} e_i) - \sum_{a=1}^{m-1} \sum_{b=a+1}^m (dt_{ab}^{re} e_a e_b) \right] \quad (2a)$$

$$DC^{de}(S) = w^{en} \left[\sum_{j=1}^n \sum_{i=1}^m (dt_{ij}^{de} s_{ij}) \right] \quad (2b)$$

In the case of an underdesigned platform the development time of the platform is expected to be shorter, resulting in lower platform development costs. However, the customisation development costs will generally be higher as the design time to derive product j from platform i will be longer. Similarly, when only a few platforms are introduced, platform development costs may be lower, but these platforms will be less customised to specific product needs, and the design time for developing products from these platforms will usually be higher.

3.3.2 Purchasing and ordering costs

An important driver of product-platform decisions is the purchasing cost. More commonality in platforms may reduce their manufacturing cost for the supplier as a result of sharing common production facilities. Due to the aggregation in demand, the buying quantities will also increase, and a price discount may be received from the supplier. The magnitude of this price discount is related to the scale benefits in manufacturing of the supplier. Whereas the costs of equipment, tooling and facilities at the supplier's side are reflected in the purchase price, the supplier's costs of packaging, transportation and distribution are included in the ordering cost. The ordering cost also includes the cost of preparing, placing and receiving goods (platforms and components) purchased from suppliers. Ordering and purchasing costs are often considered to be the most impacted by economies of scale evoked by the use of platforms.

We assume a fixed cost k incurred for each order placed. The number of orders placed per year depends on the annual demand of platforms (d_i^{pl}) and components (d_c^{co}), which are both

dependent on the product-platform decision S and the demand of the final products D^{pr} , as well as the effective buying quantities for the platforms (q_i^{pl}) and components (q_c^{co}). The unit purchase price of the platforms and components are respectively expressed by pp_i^{pl} and pp_c^{co} . As the purchasing and ordering costs are recurring yearly, we discount them over the expected lifetime of the platforms using the company's WACC. This leads to the following expressions for the purchasing and ordering costs:

$$PC^{pl}(S) = \sum_{t=0}^{life} \left(\frac{1}{(1+WACC)^t} \right) \sum_{i=1}^m \left(d_i^{pl} pp_i^{pl} + \frac{k d_i^{pl}}{q_i^{pl}} \right) \quad (3a)$$

$$PC^{co}(S) = \sum_{t=0}^{life} \left(\frac{1}{(1+WACC)^t} \right) \sum_{c=1}^{c^{max}} \left(d_c^{co} pp_c^{co} + \frac{k d_c^{co}}{q_c^{co}} \right) \quad (3b)$$

with

$$d_i^{pl} = \sum_{j=1}^n (s_{ij} \sum_{r=1}^{12} d_{rj}^{pr}) \quad \text{and} \quad d_c^{co} = s_{ij} \sum_{r=1}^{12} (d_{rj}^{pr} co_{cij}^{de})$$

The unit purchase price of the platforms and components itself depends on the effective buying quantities, as e.g. more commonality (in platforms or components) leads to larger buying quantities, resulting in better price discounts. We assume that when the purchasing quantity surpasses a certain price break, the unit purchase price decreases (known as an 'all-unit discount'). We use the algorithm described by Muckstadt and Sapra (2010, pp. 33-36) to take into account the interdependence of the purchasing quantity and price to determine the optimal buying quantity and corresponding purchase price. The effective buying quantity might be equal to the economic order quantity (EOQ) or the purchasing quantity corresponding to a certain price break. Note that when the supplier imposes a minimum order quantity (MOQ) higher than the EOQ, the EOQ is overruled and total costs are expected to increase (Porrás & Dekker, 2006).

A lower number of platforms, leading to more platform commonality, will likely result in a higher ordering quantity and thus lower platform ordering and purchasing costs, thanks to scale economies (more aggregated demand). The more oversized a platform, the higher its purchase price, which leads to higher platform ordering and purchasing costs. However, this cost increase is

partially offset by the fact that we need fewer components during customisation, leading to lower customisation ordering and purchasing costs.

3.3.3 *Inventory related costs*

Inventory related costs corresponding to product-platform decisions consist of the yearly costs connected to cycle and safety inventory, and inventory shortages for both platforms and components.

As inventories represent a need for working capital, we express the inventory holding cost as a percentage h of the platform and component purchase price (respectively pp_i^{pl} and pp_c^{co}). The holding cost also includes, among other things, the cost of storage, warehousing and insurance. We take into account the cost of obsolescence by an additional obsolescence rate x as a rate of inventory holding cost (with $x \geq 1$). The obsolescence rate is higher when platforms and components have short life-cycles or when they are highly perishable (leading to higher costs of inventory). This results in an inventory holding cost of $h \times$ times the purchase price (Joglekar & Lee, 1993).

The *cycle inventory* levels of platforms and components are affected by the platforms' and components' effective buying quantities (q_i^{pl} and q_i^{co}). Fewer platforms will reduce the total cycle inventories across platforms. However, when platforms are underdesigned, more components are needed for customisation, potentially leading to less standardisation and economies of scale on the level of components, and higher levels of component inventory.

The *safety inventory* levels of platforms and components are driven by the desired service level (sl) and the standard deviation of the forecast errors of demand over the replenishment lead time for respectively platforms (σ_i^{pl}) and components (σ_c^{co}). These forecast errors depend on the deviation between actual and forecasted monthly demand expressed by the monthly MAD (Mean Absolute Deviations) (mad_i^{pl}, mad_c^{co}) and the replenishment lead-times (lt_i^{pl}, lt_c^{co}), i.e. the time between ordering and receiving (in our model we have assumed constant lead times, although variability in lead times and uncertainty at the supplier side can be easily taken into account). The use of platforms

can lead to safety stock reductions due to their potential pooling effects (the aggregation of demand may result in more accurate demand forecasts) (Chopra & Meindl, 2007).

The expected *shortage cost* of platforms and components depends on their anticipated number of replenishment cycles per year (respectively $\frac{d_i^{pl}}{q_i^{pl}}$ and $\frac{d_c^{co}}{q_c^{co}}$) and their estimated shortages per replenishment cycle (which in turn depends on σ_i^{pl} , σ_c^{co} and the loss function $L(z) = f_s(z) - z(1 - F_s(z))$, with f_s the standard normal density function, F_s the standard normal cumulative distribution function and $z = F_s^{-1}(sl)$). The unit shortage cost of platforms and components (sc_i^{pl} and sc_c^{co}) reflects the loss in margin and goodwill in case of a stock-out. In theory, this shortage cost should also impact the company's service levels (Silver, et al., 1998, p. 263). In practice, however, many companies use identical service levels for all products.

We follow the common assumptions that forecast errors are normally distributed and not biased (Jacobs & Wagner, 1989). This leads to the following (discounted) platform and customisation inventory related costs (Silver, et al., 1998, pp. 253-301):

$$IC^{pl}(S) = \sum_{t=0}^{life} \left(\frac{1}{(1+WACC)^t} \right) \sum_{i=1}^m \left(\frac{xh pp_i^{pl} q_i^{pl}}{2} + xh z pp_i^{pl} \sigma_i^{pl} + \frac{L(z) sc_i^{pl} \sigma_i^{pl} d_i^{pl}}{q_i^{pl}} \right) \quad (4a)$$

$$IC^{co}(S) = \sum_{t=0}^{life} \left(\frac{1}{(1+WACC)^t} \right) \sum_{c=1}^{c^{max}} \left(\frac{xh pp_c^{co} q_c^{co}}{2} + xh z pp_c^{co} \sigma_c^{co} + \frac{L(z) sc_c^{co} \sigma_c^{co} d_c^{co}}{q_c^{co}} \right) \quad (4b)$$

with $z = F_s^{-1}(sl)$

$$L(z) = f_s(z) - z(1 - F_s(z))$$

$$\sigma_i^{pl} = 1.25 \sqrt{lt_i^{pl}} mad_i^{pl}$$

$$\sigma_c^{co} = 1.25 \sqrt{lt_c^{co}} mad_c^{co}$$

$$mad_i^{pl} = \frac{\sum_{r=1}^{12} \left| \sum_{j=1}^n (s_{ij} (d_{rj}^{pr} - \hat{d}_{rj}^{pr})) \right|}{12}$$

$$mad_c^{co} = \frac{\sum_{r=1}^{12} \left| \sum_{i=1}^m \sum_{j=1}^n \left(s_{ij} co_{cij}^{de} (d_{rj}^{pr} - \hat{d}_{rj}^{pr}) \right) \right|}{12}$$

3.3.4 Transformation costs of platforms to final products

During the transformation phase of platforms into final products a typical final processing cost is involved. This happens in the operational phase, which differs from the development phase (discussed in section 3.3.1). Here, we do not consider the material costs of components needed during customisation (as these are already included in the purchasing costs), but we look at the effort and time (tt_{ij}^{de}) needed by the operators (with wage w^{op}) to transform a (common) platform i into an (individual) final product j . The annual demand of products derived from a specific platform (D^{de}) is derived from the product-platform scenario S .

$$TC^{de}(S) = \sum_{t=0}^{life} \left(\frac{1}{(1+WACC)^t} \right) \sum_{j=1}^n \sum_{i=1}^m \left(w^{op} d_{ij}^{de} tt_{ij}^{de} \right) \quad (5a)$$

with

$$d_{ij}^{de} = s_{ij} \sum_{r=1}^{12} d_{rj}^{pr}$$

In case of underdesigned platforms or a small number of platforms, the transformation costs are expected to be higher than in the case of overdesign or many platforms, since more adaptations are to be done and/or more different components to be added.

4 Trade-off underlying the optimal product-platform decisions

In essence, the supply chain costs that are impacted by product-platform decisions boil down to supply chain costs related to the platforms (Platform Costs $C^{platform}(S)$) and to the customisation to final products (Customisation Costs $C^{customisation}(S)$) (see Figure 2), with

$$C^{platform}(S) = DC^{pl}(S) + PC^{pl}(S) + IC^{pl}(S)$$

$$C^{customisation}(S) = DC^{de}(S) + PC^{co}(S) + IC^{co}(S) + TC^{de}(S)$$

Moreover, an increase in platform costs usually implies a decrease in customisation costs, and vice versa. The two decision types driving this cost trade-off are:

- (1) whether there are few or many platforms introduced (i.e. the number of platforms);
- (2) whether the platforms are under- or overdesigned (which depends on the decision about which platforms are developed and which products are derived from which platforms).

Indeed, a higher number of platforms, as well as more overdesigned platforms simultaneously lead to an increase in platforms cost and a decrease in customisation cost. The opposite holds in case of a small number of platforms, or underdesigned platforms. This leads us to formulate the following lemma, which turns out to be a useful guideline in making product-platform decisions:

Lemma 1: Adding an extra platform reduces total supply chain costs as long as (with x being the current number of platforms):

$$\frac{C^{platform}(x+1) - C^{platform}(x)}{C^{customisation}(x) - C^{customisation}(x+1)} < 1.$$

Lemma 1 implies that an extra platform should be added when the cost increase related to platforms is smaller than the cost reductions related to customisation (i.e. deriving final products from platforms). Although the lemma is intuitively logical, it emphasizes how the optimal product-platform decisions depend on the magnitude of the trade-off. Claiming that one platform, multiple platforms or the absence of platforms will under any circumstance lead to a better solution (i.e. in terms of less supply chain costs) is impossible, since the optimal number of platforms is the result of quantifying the overall supply chain costs in an integrated way. The lemma also proves to be a helpful guideline in making the optimal product-platform decisions in a real-business context. In what follows we show the applicability of our model in a business context.

5 Numerical experiment

We applied our quantitative cost model on a global technology company to determine how many and which platforms should be developed, and which products should be derived from which platforms.

The global technology company develops and produces medical screens that are used in the healthcare industry for visualisation purposes, for example by specialists who apply them as a visualisation support during endoscopy or surgery. The portfolio of medical screens of the company is increasing every year, mainly driven by customer requests and technological evolution. Currently, there are 27 different versions of medical screens (seen as the final products), which are derived from a set of 15 different printed circuit boards (seen as the platforms). Extra components are added to customise the platforms to product variants. The R&D department of the company is responsible of the development of the printed circuit boards, whereas the production is outsourced. A medical screen is derived from a printed circuit board-platform by adding the following component types: connectors (such as screws), an LCD panel, (electro)-mechanical components, wires and cables, optical components, and miscellaneous components (such as packaging and plastic parts). Various components are needed during customisation, depending on which kind of platform a certain product is derived from.

For reasons of illustration and confidentiality of company information, we narrow down the scope of the case to 12 final products that can (possibly) be derived from 12 platforms (the complexity of the decision making remains, despite the reduction). The medical screens and their associated platforms vary in terms of low (L) or high (H) image quality, colour (C) or grey (G) scale and the amount of MPs (megapixels).

Table 1 lists the set of screens considered in our case according to their level of complexity and functionality, which means product 1 is the lowest end product and product 12 is the highest end product. The same applies to platforms, where platform 1 is the lowest end platform and platform 12 is the highest end platform. Hence, when using a platform with a number that is higher than the product number, the platform is overdesigned (e.g. platform 6 is overdesigned for the functionality requirements of product 1). For this case, inputs were obtained by extracting the Bill of Material (BOM) of all products considered (which includes the components needed, the components' prices, the platforms used and the platforms' prices), by input of two R&D managers on the expected time needed to develop platforms and products from those platforms, by interviews with supply chain

managers on the holding cost and ordering cost applicable in the company, and by the analysis of purchasing contracts in which quantity discounts and purchasing costs are elaborated on. Note that the development time can be reduced thanks to economies of scope of developers learning from the elaboration of other platforms, and the purchase price can be reduced thanks to quantity discounts (for which we assume purchase prices to be concave decreasing as a function of the order quantities).

Finally, Table 1 provides the annual demand and its monthly forecast accuracy, expressed by its mean absolute deviation (MAD), of the 12 different products (4MP screens are less requested by customers and product 12 has a lower demand and less accurate forecasts as it is a new product introduction).

(Please insert Table 1 here)

Technically, the high quality products (i.e. product 6 to 12) cannot be derived from low quality platforms (platform 1 to 5). The reason for this is that the low quality screens have already existed within the company for many years and that they have an outdated printed-circuit board platform architecture. However, it is possible to use the newer high quality platforms to make the lower quality products (except for product 12, a 10 MP screen, which is very specific and innovative in design and needs its own dedicated platform 12 which cannot (yet) be used for the derivation of the other screens). In the base case scenario (where each product is derived from its dedicated platform) the low-end platforms are used for the low-end products, the high-end products are derived from high-end platforms, and customisation during the design and processing phase is minimal.

Table 2 gives an overview of the values of the company input parameters used in our numerical experiment (we have assumed lead times of all platforms to be the same and constant, as well as lead time of all components, but this should not necessarily be the case).

(Please insert Table 2 here)

5.1. *Solution algorithm*

Our model's architecture is scalable and customisable to a random number of products and platforms. However, for large problem instances, explicit enumeration can be very time-consuming as we face an NP-hard problem (due to the integer optimisation with the decision variable S containing 0-1 elements (Karp, 1972)).

To solve large problem instances, we need a heuristic algorithm to obtain results within reasonable computation time. We suggest the use of a simulated annealing algorithm (SA) (Kirkpatrick, et al., 1983), of which the pseudo-code is provided in Figure 3.

The first step in the algorithm reads the company's inputs, selects an initial scenario S_{start} , and initialises the level of the temperature, the number of iterations and steps ($Temp \leftarrow 1$, $Iteration \leftarrow 0$ and $Step \leftarrow 0$). We set S_{start} equal to the base case scenario for which each product is derived from its own dedicated platform and S is a diagonal matrix. The scenario S_{start} and its supply chain costs (calculated with the function 'Cost') are the starting point for the algorithm: S_{start} is set equal to $S_{current}$ and S_{best} , while its supply chain costs are set equal to $C_{current}$ and C_{best} .

The second step of the algorithm constructs a neighbouring scenario S_{new} , by repositioning a randomly selected product a from $S_{current}$ to another technically feasible randomly selected platform b .

In the third step, the total supply chain costs associated with S_{new} (C_{new}) are compared to the lowest cost found up till now (C_{best}). Either $C_{new} < C_{best}$, in which case we replace S_{best} and $S_{current}$ by S_{new} , and C_{best} and $C_{current}$ by C_{new} ; or $C_{new} \geq C_{best}$, in which case there is a positive probability ($rand < Temp$) that $S_{current}$ is replaced by S_{new} . The higher the temperature, the more likely $S_{current}$ is replaced by a worse scenario S_{new} , in the hope of exploring a new part of the solution landscape that contains better results.

In the last step of the algorithm, temperature is cooled down linearly, depending on the value of $Step$ and $Step_{max}$. Note that we have not chosen for a more complex probability calculation such as the Metropolisian acceptance probability (Sadeghi, et al., 2011), and that we do not allow reheating the temperature.

As long as the temperature exceeds zero and Iteration is at most equal to Iterationmax, phases two till four of the algorithm are repeated

(Please insert Figure 3 about here)

Table 3 provides computation times for a number of scenarios using explicit enumeration, as well as computation times and optimality gaps using simulated annealing (we use a 3.47 GHz with 8 GB RAM, and we coded our problem in MATLAB R2013b). It shows that the proposed SA algorithm provides relatively quick solutions to large-scale problems. The (relative) optimality gap, represented by the relative difference between the lowest cost obtained using SA and the optimal cost (obtained using explicit enumeration), decreases fast as the SA runs for a longer time (i.e. the results converge fast to the optimal solution).

(Please insert Table 3 about here)

5.2. Results

Figure 4 shows the results for our numerical case study. It displays the supply chain costs for the different scenarios where gradually more platforms are introduced. For each scenario, we solve which platforms should be introduced and which products should be produced on each of the platforms to generate the lowest supply chain costs. This information is provided in the table below the plot. The plot itself visualises how the supply chain costs change as a function of the number and type of platforms used and the decision about which products are derived from which platforms, and how it is determined by the underlying trade-off between the platform and customisation costs. In the business case of the medical screen producer, the optimal product-platform decision is to introduce 9 platforms and derive product 1 and product 3 from platform 1; product 6, 9 and 11 from platform 6; and the other products from their own dedicated platform. Observe that the ratio of lemma 1 exceeds one as soon as the number of platforms exceeds 9, which implies that selecting a scenario with an

additional platform is not attractive any more. Compared to the base case scenario of introducing 12 platforms (i.e. each product is derived from its own perfectly matching platform), we find that introducing (common) platforms reduces total supply chain costs. However, maximizing platform commonality by only having two or three platforms from which all products are derived, in this case does not lead to the lowest supply chain costs. This can be explained by the high customisation costs in case of high platform commonality, which spurs from the large diversity between the 12 products offered.

The analysis also reveals that total supply chain costs only differ slightly whether 6, 7, 8, 9 or 10 platforms are used. This is due to the fact that not only the number of platforms is driving total supply chain costs, but also the decision about which products to allocate to which platforms and the type of platforms developed. Comparing the scenarios from 6 to 10 platforms, the decision to develop products 1, 4, 6, 7, 10 and 12 from their own dedicated platforms and to derive products 6, 9 and 11 from platform 6 remains unchanged. Given the sensitivity of our model to input inaccuracies, it is reassuring to see that these platform development and product-platform allocation decisions remain robust. The decision whether to develop product 2, 3, 5 or 8 from a dedicated platform turns out to have much less impact on total supply chain costs.

(Please insert Figure 4 about here)

5.3. Sensitivity analysis

Since the product-platform decisions are driven by the model's input parameters, which are susceptible to mistakes, misjudgement or changes, we show how our model can easily be used to conduct a sensitivity analysis (see results Figure 5).

- For instance, if platform development time would be 10 per cent longer (compared to Table 1, and keeping the other input parameters constant), then the optimal number of platforms remains 9 and product-platform allocation decisions do not change. However, in case platform development time would be 25 per cent longer, the optimal number of platforms

decreases to 6, and products 1, 2, 3 and 5 are derived from platform 1; products 6, 8, 9 and 11 are derived from platform 6; and the other products are derived from their own dedicated platform. An increase in platform development time leads to the decision to use less platforms or to use more underdesigned platforms, due to the increased platform costs in relation to the customisation costs. Similarly, a change in development time can impact the decision about which platforms to develop and which products are derived from them (which can be seen for instance in the scenario of using 3 platforms). Our model can easily quantify the impact of these parameter changes.

- In case of higher demand, the commonality benefits in purchasing, ordering and inventory are likely to compensate the high platform development costs (as there are more economies of scale). Also, the higher the demand, the higher the customisation costs. This is in favour of introducing more platforms. If in our example demand of all products doubles (see Table 1, and assuming other parameters constant), and assuming demand forecasts are also twice as high, we find that the optimal decision is to develop 10 platforms instead of 9, in which case products 2 and 3 are derived from platform 2; products 6 and 9 are derived from platform 6; and the other products are derived from their dedicated platform. Also here the product-platform allocation decisions may change.
- We can also quantify the impact of quantity discounts. If the company would receive higher quantity discounts in the purchase of platforms from its suppliers, then the platform costs decrease relative to the customisation costs, which may lead to more platforms. In our case, a discount of 25 per cent instead of 5 per cent (for $50 \leq Q < 100$) does not impact the optimal number of platforms (it still equals 9), but the product-platform allocation decisions change (e.g. product 11 is now derived from platform 10 instead of from platform 6).
- Finally, we simulated the impact of a change in lifetime. For example, when the lifetime doubles the optimal number of product platforms remains 9, but when the platform lifetime is three times longer, then the optimal number of platforms would increase to 10. For higher platform lifetimes, the high investment in platform development is compensated by a longer

period of recurring purchasing, inventory and transformation costs on the level of platforms and customisation. Note that a similar reasoning holds for the WACC for which a higher capital cost may lead to a decrease in the optimal number of platforms.

6 Conclusions

In this paper we have presented a model to facilitate product-platform decisions by quantifying their impact on the total supply chain costs. The model enables to evaluate how many and which platforms should be developed, and which products should be derived from which platforms. This contributes to existing literature on platforms that mostly focuses on other decisions, such as optimising the design and architecture of products, platforms and the supply chain. The model is unique in the scope of the considered supply chain cost families – namely the cost families relating to development, purchasing, inventory management and processing to transform platforms into final products – as well as the way how these costs are quantified. Future research could be devoted to costs arising after the final products have been manufactured, such as the cost of marketing and sales, product maintenance, product installation and end-of-life management (e.g. take-back and recovery).

Our model reveals the cost trade-off when making product-platform decisions: the costs related to platforms versus the costs related to the customisation of platforms to final products. This trade-off is driven by the number of platforms developed and the decision to “under- or overdesign” platforms (compared to the products derived from it). We applied our model to a global technology company with the intention to validate its practical use. Dealing with a complex model, big datasets and an increasing number of product-platform scenarios to evaluate, we presented a simulated annealing heuristic to solve the problem time-efficiently. Overall we can conclude that, contrary to what is commonly prompted in literature, product-platform decisions can only be made when considering all supply chain costs in an integrated way such as we have done in our model.

References

- Agard, B. & Bassetto, S., 2013. Modular design of product families for quality and cost. *International Journal of Production Research*, 51(6), pp. 1648-1667.
- Agrawal, T. et al., 2013. A hybrid model of component sharing and platform modularity for optimal product family design. *International Journal of Production Research*, 51(2), pp. 614-625.
- Baud-Lavigne, B., Agard, B. & Penz, B., 2012. Mutual impacts of product standardization and supply chain design. *International Journal of Production Economics*, Volume 135, pp. 50-60.
- Ben-Arieh, D., Easton, T. & Choubey, A., 2009. Solving the multiple platforms configuration problem. *International Journal of Production Research*, 47(7), pp. 1969-1988.
- Chen, C. & Wang, L., 2008. Product platform design through clustering analysis and information theoretical approach. *International Journal of Production Research*, 46(15), pp. 4259-4284.
- Chopra, S. & Meindl, P., 2007. *Supply Chain Management: Strategy, Planning, & Operation*. 3 ed. New Jersey: Pearson Prentice Hall.
- Collier, D. A., 1981. The measurement and operating benefits of component part commonality. *Decision Sciences*, 12(1), pp. 85-96.
- de Weck, O. L., Suh, E. S. & Chang, D., 2003. *Product family and platform portfolio optimization*. Chicago, ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference.
- Deflem, Y. & Van Nieuwenhuyse, I., 2013. Managing inventories with one-way substitution: A newsvendor analysis. *European Journal of Operational Research*, 228(3), pp. 484-493.
- D'Souza, B. & Simpson, T. W., 2003. A genetic algorithm based method for product family design optimization. *Engineering Optimization*, 35(1), pp. 1-18.
- Du, G., Jiao, R. J. & Chen, M., 2014. Joint optimization of product family configuration and scaling design by Stackelberg game. *European Journal of Operational Research*, 232(2), pp. 330-341.
- Farrell, R. S. & Simpson, T. W., 2010. Improving cost effectiveness in an existing product line using component product platforms. *International Journal of Production Research*, 48(11), pp. 3299-3317.

- Fellini, R., Kokkolaras, M. & Papalambros, P. Y., 2006. Quantitative platform selection in optimal design of product families, with application to automotive engine design. *Journal of Engineering Design*, 17(5), pp. 429-446.
- Gonzalez-Zugasti, J. P., Otto, K. N. & Baker, J. D., 2001. Assessing value in platformed product family design. *Research in Engineering Design*, 13(1), pp. 30-41.
- Holttä, K., Suh, E. S. & de Weck, O., 2005. *Tradeoff between modularity and performance for engineered systems and products*. Melbourne, International Conference on Engineering Design.
- Jacobs, R. A. & Wagner, H. M., 1989. Reducing inventory system costs by using robust demand estimators. *Management Science*, 35(7), pp. 771-787.
- Jiao, J. R., Simpson, T. W. & Siddique, Z., 2007. Product family design and platform-based product development: a state-of-the-art review. *Journal of Intelligent Manufacturing*, 18(1), pp. 5-29.
- Joglekar, P. & Lee, P., 1993. An exact formulation of inventory costs and optimal lot size in face of sudden obsolescence. *Operations Research Letters*, 14(5), pp. 283-290.
- Karp, R. M., 1972. *Reducibility among combinatorial problems*. s.l.:Springer US.
- Khalaf, R. E. H., Agard, B. & Penz, B., 2011. Simultaneous design of a product family and its related supply chain using a Tabu Search algorithm. *International Journal of Production Research*, 49(19), pp. 5637-5656.
- Kirkpatrick, S., Gelatt, C. & Vecchi, M., 1983. Optimization by Simulated Annealing. *Science*, 220(1), pp. 671-680.
- Krishnan, V. & Gupta, S., 2001. Appropriateness and impact of platform-based product development. *Management Science*, 47(1), pp. 52-68.
- Meyer, M. H. & Lehnerd, A. P., 1997. *The power of product platforms: building value and cost leadership*. New York: The Free Press.
- Muckstadt, J. A. & Sapra, A., 2010. *Principles of inventory management*. New York: Springer.
- Muffatto, M., 1999. Introducing a platform strategy in product development. *International Journal of Production Economics*, Volume 60-61, pp. 145-153.

- Porrás, E. & Dekker, R., 2006. An efficient optimal solution method for the joint replenishment problem with minimum order quantities. *European Journal of Operational Research*, 174(3), pp. 1595-1615.
- Robertson, D. & Ulrich, K., 1998. Planning for product platforms. *Sloan management review*, 39(4), pp. 19-31.
- Sadeghi, A., Alem-Tabriz, A. & Zandieh, M., 2011. Product portfolio planning: a metaheuristic-based simulated annealing algorithm. *International Journal of Production Research*, 49(8), pp. 2327-2350.
- Seepersad, C. C., Hernandez, G. & Allen, J. K., 2000. *A quantitative approach to determining product platform extent*. Baltimore, ASME Design Engineering Technical Conference.
- Silver, E. A., Pyke, D. F. & Peterson, R., 1998. *Inventory Management and Production Planning and Scheduling*. 1 ed. New York: John Wiley & sons.
- Simpson, T. W., Jiao, J. R., Siddique, Z. & Hölttä-Otto, K., 2014. *Advances in Product Family and Product Platform Design: Methods & Applications*. 1 ed. New York: Springer.
- Thyssen, J., Israelsen, P. & Jorgensen, B., 2006. Activity-based costing as a method for assessing the economics of modularization - A case study and beyond. *International Journal of Production Economics*, 103(1), pp. 252-270.
- Yadav, S. R., Mishra, N., Kumar, V. & Tirwari, M., 2011. A framework for designing robust supply chains considering product development issues. *International Journal of Production Research*, 49(20), pp. 6065-6088.
- Zhang, X. & Huang, G. Q., 2010. Game-theoretic approach to simultaneous configuration of platform products and supply chains with one manufacturing firm and multiple cooperative suppliers. *International Journal of Production Economics*, 124(1), pp. 121-136.
- Zhang, X., Huang, G. Q., Humphreys, P. K. & Botta-Genoulaz, V., 2010. Simultaneous configuration of platform products and manufacturing supply chains: comparative investigation into impacts of different supply chain coordination schemes. *Production Planning & Control: The Management of Operations*, 21(6), pp. 609-627.

Appendix 1: Overview of the notations used

The grey shaded boxes indicate the inputs needed from the firm, whereas the white boxes are decision variables or variables derived from the inputs.

	<i>Element or number</i>	<i>Vectors and matrices</i>	<i>Size of vectors and matrices</i>	<i>Interpretation</i>
Decision variables	s_{ij}	S	(m, n)	$s_{ij} = 1$ if product j is derived from platform i and 0 otherwise
	e_i	E	$(m, 1)$	$e_i = 1$ if platform i is developed and 0 otherwise
Technical constraint	t_{ij}	T	(m, n)	$t_{ij} = 1$ if it is (technically) possible to derive product j from platform i and 0 otherwise
General input parameters	m			Maximum number of platforms
	n			Number of products delivered to the market
	c^{max}			Maximum number of components
	q_1			Number of price breaks for the purchase price of platforms
	q_2			Number of price breaks for the purchase price of components
	$life$			Average of expected lifetime of the different platforms (in years)
	$WACC$			Annual interest rate (as a percentage)(value between 0 and 1) which equals the Weighted Average Cost of Capital
Input parameters for development costs	dt_i^{pl}	DT^{pl}	$(m, 1)$	Design time of platform i (in hours)
	dt_{ij}^{de}	DT^{de}	(m, n)	Design time, on top of the platform design time, needed to derive product j from platform i (in hours)
	dt_{ab}^{re}	DT^{re}	(m, m)	Design time reduction (in hours) if both platform a and b are developed together, for which the elements equal zero when $b \geq a$
	w^{en}			Wage of an engineer (in euro per hour)
Input parameters for ordering and purchasing costs	k			Fixed cost per order (in euro per order)
	$pp_i^{pl, pb}$	$PC^{pl, pb}$	(m, q_1)	Purchase price (in euro per unit) of platform i given a purchasing quantity equal or higher than the buying quantity corresponding to price break pb and smaller than the buying quantity of the next highest price break $pb+1$
	pp_i^{pl}	PC^{pl}	$(m, 1)$	Purchase price (in euro per unit) of platform i given the optimal purchasing quantity q_i^{pl}
	$pp_c^{co, pb}$	$PC^{co, pb}$	(c^{max}, q_2)	Purchase price (in euro per unit) of component c given a purchasing quantity equal or higher than the buying quantity corresponding to price break pb and smaller than the buying quantity of the next highest price break $pb+1$
	pp_c^{co}	PC^{co}	$(c^{max}, 1)$	Purchase price (in euro per unit) of component c given the optimal purchasing quantity q_c^{co}
	co_{cij}^{de}	CO^{de}	(c^{max}, m, n)	Amount (in units) of component c needed to derive product j from platform i
Input parameters for product, platform and component demand	d_{rj}^{pr}	D^{pr}	$(12, n)$	Actual demand (in units) of product j in month r , for which D^{pr} covers 1 year
	\hat{d}_{rj}^{pr}	\hat{D}^{pr}	$(12, n)$	Forecasted demand (in units) of product j in month r , for which \hat{D}^{pr} covers 1 year
	d_i^{pl}	D^{pl}	$(m, 1)$	Actual annual demand (in units) of platform i

	d_c^{co}	D^{co}	$(c^{max}, 1)$	Actual annual demand (in units) of component c
	d_{ij}^{de}	D^{de}	(m, n)	Actual annual demand (in units) of product j derived from platform i
Inputs parameters for inventory related costs (including the cost of cycle and safety inventory, and inventory shortages)	h			Inventory holding cost (as a percentage of purchase price) ($0 \leq h \leq 1$)
	x			Obsolescence cost (as a percentage of holding cost) ($x \leq 1$)
	sl			Desired service level (as a percentage of customers that do not experience a stock out) ($0 \leq sl \leq 1$)
	z			Safety factor
	$L(z)$			Loss function
	sc_i^{pl}			Unit shortage cost of platform i (in euro per unit)
	sc_c^{co}			Unit shortage cost of component c (in euro per unit)
	lt_i^{pl}	LT^{pl}	$(m, 1)$	Average replenishing lead time of a platform i (in hours)
	lt_c^{co}	LT^{co}	$(c^{max}, 1)$	Average replenishing lead time of component c (in hours)
	mad_i^{pl}	MAD^{pl}	$(m, 1)$	Mean absolute deviation of platform i
	mad_c^{co}	MAD^{co}	$(c^{max}, 1)$	Mean absolute deviation of component c
	σ_i^{pl}	σ^{pl}	$(m, 1)$	Standard deviation of the errors of forecasts of demand of platform i over the replenishment lead time
	σ_c^{co}	σ^{co}	$(c^{max}, 1)$	Standard deviation of the errors of forecasts of demand of product j over the replenishment lead time
	moq_i^{pl}	MOQ^{pl}	$(m, 1)$	Minimum order quantity of platform i
	moq_c^{co}	MOQ^{co}	$(c^{max}, 1)$	Minimum order quantity of component c
Input parameters for transformation costs	q_i^{pl}	Q^{pl}	$(m, 1)$	(Optimal) purchasing quantity of platform i
	q_c^{co}	Q^{co}	$(c^{max}, 1)$	(Optimal) purchasing quantity of component c
	tt_{ij}^{de}	TT^{de}	(m, n)	Time (in hours) needed to transform platform i to product j
	w^{op}			Wage of an operator (in euro per hour)
Categorisation of costs	$C^{platform}(S)$			(Discounted) platform costs
	$C^{customisation}(S)$			(Discounted) customisation costs
	$DC^{pl}(S)$			Platform development costs
	$DC^{de}(S)$			Customisation development costs needed to derive final products from platforms during development
	$PC^{pl}(S)$			(Discounted) platform purchasing costs
	$PC^{co}(S)$			(Discounted) purchasing costs of components needed for customisation
	$IC^{pl}(S)$			(Discounted) platform inventory related costs (including the cost of cycle and safety inventory cost, and inventory shortages)

	$IC^{co}(S)$	(Discounted) inventory related costs of components needed for customisation (including the cost of cycle and safety inventory, and inventory shortages)
	$TC^{de}(S)$	(Discounted) transformation costs needed to derive final products from platforms during production

Tables

Table 1: Overview of the medical screens and platforms considered in our numerical experiment

Product/ Platform number	Image quality level	Colour or grey scale	Amount of MPs	Product/ Platform name	Platform development time (in person days)	Platform purchase price (in euro per unit)	Product demand (in units per year)	Monthly mean absolute deviation (MAD) of product demand
1	Low	Grey	2	L-G-2	1200	120	222	3.83
2	Low	Colour	3	L-C-3	1750	170	268	5.00
3	Low	Colour	4	L-C-4	1900	190	50	8.00
4	Low	Grey	5	L-G-5	2200	230	517	6.75
5	Low	Grey	6	L-G-6	2400	270	314	7.83
6	High	Grey	2	H-G-2	2800	330	383	7.14
7	High	Colour	2	H-C-2	3100	530	471	6.86
8	High	Grey	3	H-G-3	3900	570	362	5.72
9	High	Colour	4	H-C-4	4400	630	112	6.33
10	High	Grey	5	H-G-5	5300	670	592	7.72
11	High	Colour	6	H-C-6	7200	690	225	5.25
12	High	Grey	10	H-G-10	8200	730	86	11.44

Table 2: Additional input parameters

WACC	Average platform lifetime	Wage engineer	Wage operator	Holding cost	Obsolescence rate	Service level	Lead time platforms	Lead time components	Cost per order	Unit shortage cost of the platforms and components	Discount scheme (discount percentage in function of the size of the order quantity Q)
10.5 %	5 years	70 euro per hour	55 euro per hour	25 % of purchase price	1.10	95%	45 working days	20 working days	125 euro per order	Equals the purchase price of the platforms and components	$50 \leq Q < 100$: 5% $100 \leq Q < 150$: 7.85% $150 \leq Q < 200$: 9.69% $200 \leq Q$: 10.60%

Table 3: Computational results of simulated annealing and explicit enumeration for different number of scenarios (computation time in seconds)

Number of products (n)	Number of platforms (m)	No of scenarios	Explicit Enumeration Computation time	Simulated Annealing		Simulated Annealing		Simulated Annealing		Simulated Annealing		Simulated Annealing	
				Computation time (Iterationmax = 1000)	Optimality gap	Computation time (Iterationmax = 5000)	Optimality gap	Computation time (Iterationmax = 10K)	Optimality gap	Computation time (Iterationmax = 50K)	Optimality gap	Computation time (Iterationmax = 500K)	Optimality gap
5	5	3 125	23,57	5,73	2,82%	30,11	0,00%	59,48	0,00%	294,69	0,00%	2934,40	0,00%
6	6	46 656	426,33	7,72	7,45%	38,22	1,32%	76,43	1,32%	387,61	0,00%	3837,60	0,00%
7	7	823 543	9321,90	9,98	28,36%	48,92	19,00%	97,79	11,85%	488,84	2,66%	4921,80	0,00%
8	8	16 777 216	245600,00	12,50	62,75%	62,39	8,34%	123,21	5,45%	613,32	4,30%	6152,70	0,33%

Figures

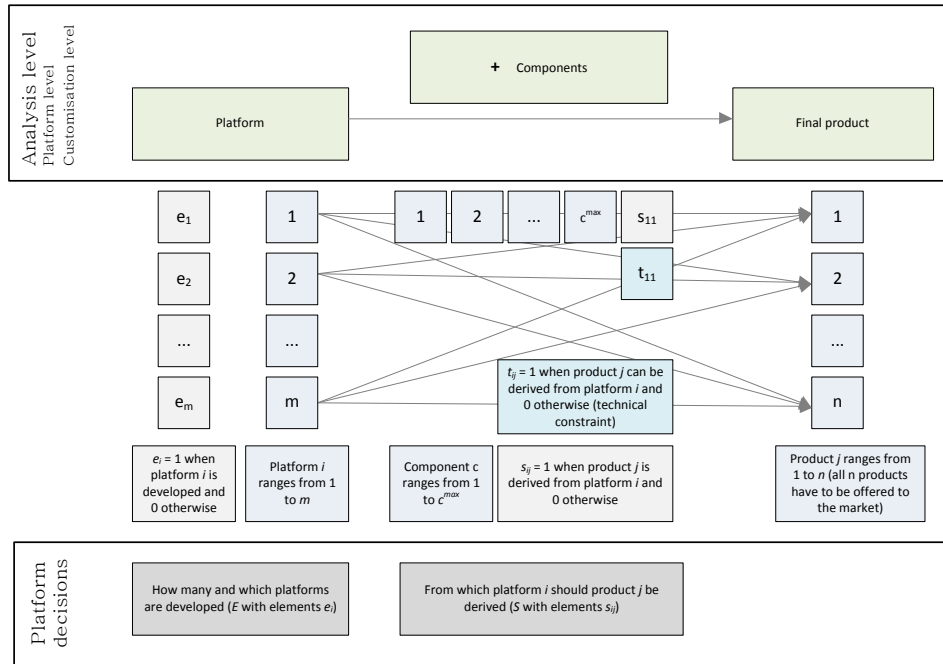


Figure 1: Illustration of our product-platform decision model

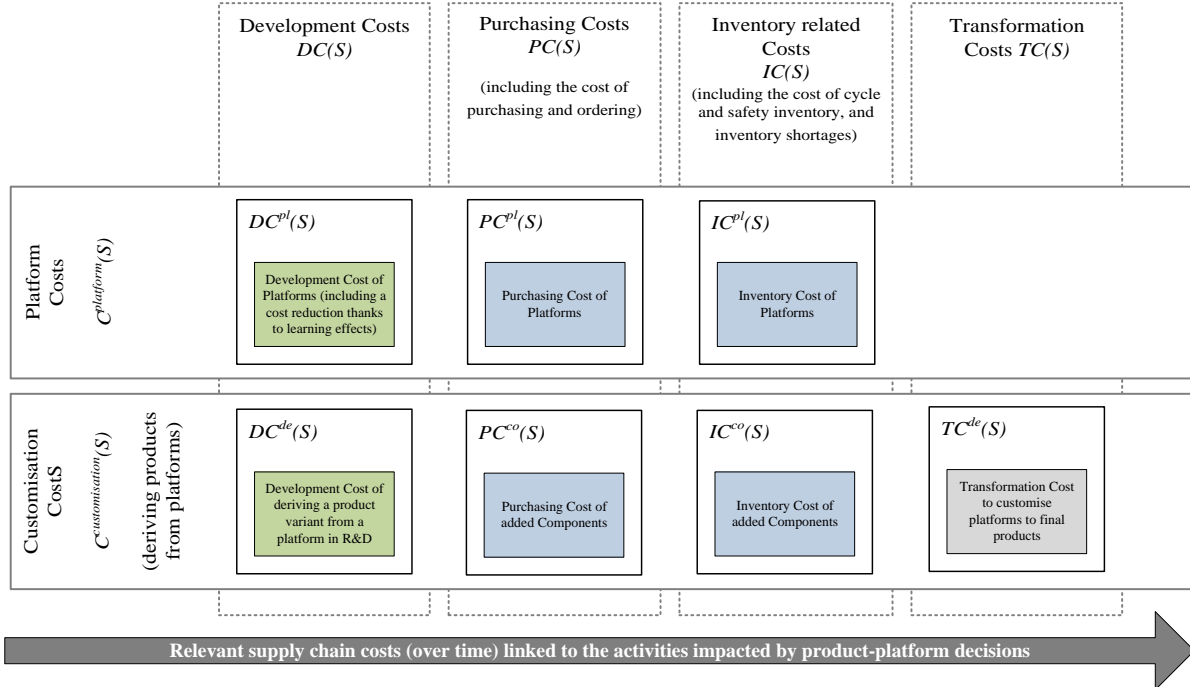


Figure 2: Overview of the supply chain costs related to product-platform decisions

Define

- 'Sbest' as the current best (most cost-efficient) product-platform scenario found during the algorithm execution, and 'Cbest' as the corresponding total supply chain cost
- 'Scurrent' as the current product-platform scenario, and 'Ccurrent' as the corresponding total supply chain cost
- 'Sstart' as the product-platform scenario from which we start the algorithm
- 'm' as the number of platforms and 'n' as the number of products
- 'S' as a matrix of size $m \times n$, representing a product-platform scenario
- 'T' as the technical feasibility matrix
- 'Temp' as the temperature level
- 'Iterationmax' as the maximum number of iterations we are willing to wait for a better solution and define 'Iteration' to keep track of the number of iterations
- 'Stepmax' as the maximum number of steps and define 'Step' to keep track of the number of steps, so that the temperature is each time reduced linearly

Consider the function $\text{Cost}(S)$ which calculates for a given product-platform scenario S the associated supply chain costs.

Step 1) Initialisation

Load company inputs

Choose the value of Stepmax and Iterationmax

Set $\text{Temp} \leftarrow 1$, $\text{Iteration} \leftarrow 0$, $\text{Step} \leftarrow 0$

$\text{Scurrent} \leftarrow \text{Sstart}$ and $\text{Ccurrent} \leftarrow \text{Cost}(\text{Sstart})$

$\text{Sbest} \leftarrow \text{Scurrent}$ and $\text{Cbest} \leftarrow \text{Ccurrent}$

Step 2) Select a neighbouring scenario Snew

While ($\text{Iteration} \leq \text{Iterationmax}$ and $\text{Temp} > 0$)

$\text{Iteration} = \text{Iteration} + 1$, $\text{Step} = \text{Step} + 1$

$a \leftarrow$ random number between 1 and n

$b \leftarrow$ random number between 1 and m

If ($T(b,a) = 1$ and $\text{Scurrent}(b,a) = 0$)

Set $\text{Snew} \leftarrow \text{Scurrent}$

Replace $\text{Snew}(:,a)$ by 0 and $\text{Snew}(b,a)$ by 1

$\text{Cnew} \leftarrow \text{Cost}(\text{Snew})$

Step 3) Evaluate the costs of the neighbouring scenario

Step 3.1) The costs of Snew are lower than those of the best scenario up till now

If $\text{Cnew} < \text{Cbest}$

Set $\text{Scurrent} \leftarrow \text{Snew}$, $\text{Ccurrent} \leftarrow \text{Cnew}$, $\text{Sbest} \leftarrow \text{Scurrent}$, $\text{Cbest} \leftarrow \text{Ccurrent}$

Step 3.2) The costs of Snew are higher than those of the best scenario up till now

Else

$p \leftarrow \text{rand}$

If $p < \text{Temp}$

Set $\text{Scurrent} \leftarrow \text{Snew}$ and $\text{Ccurrent} \leftarrow \text{Cnew}$

End If

End If

Step 4) Decrease of the temperature

$\text{Temp} = 1 - (\text{step}/\text{stepmax})$

End If

End While

Figure 3: Pseudo-code simulated annealing algorithm

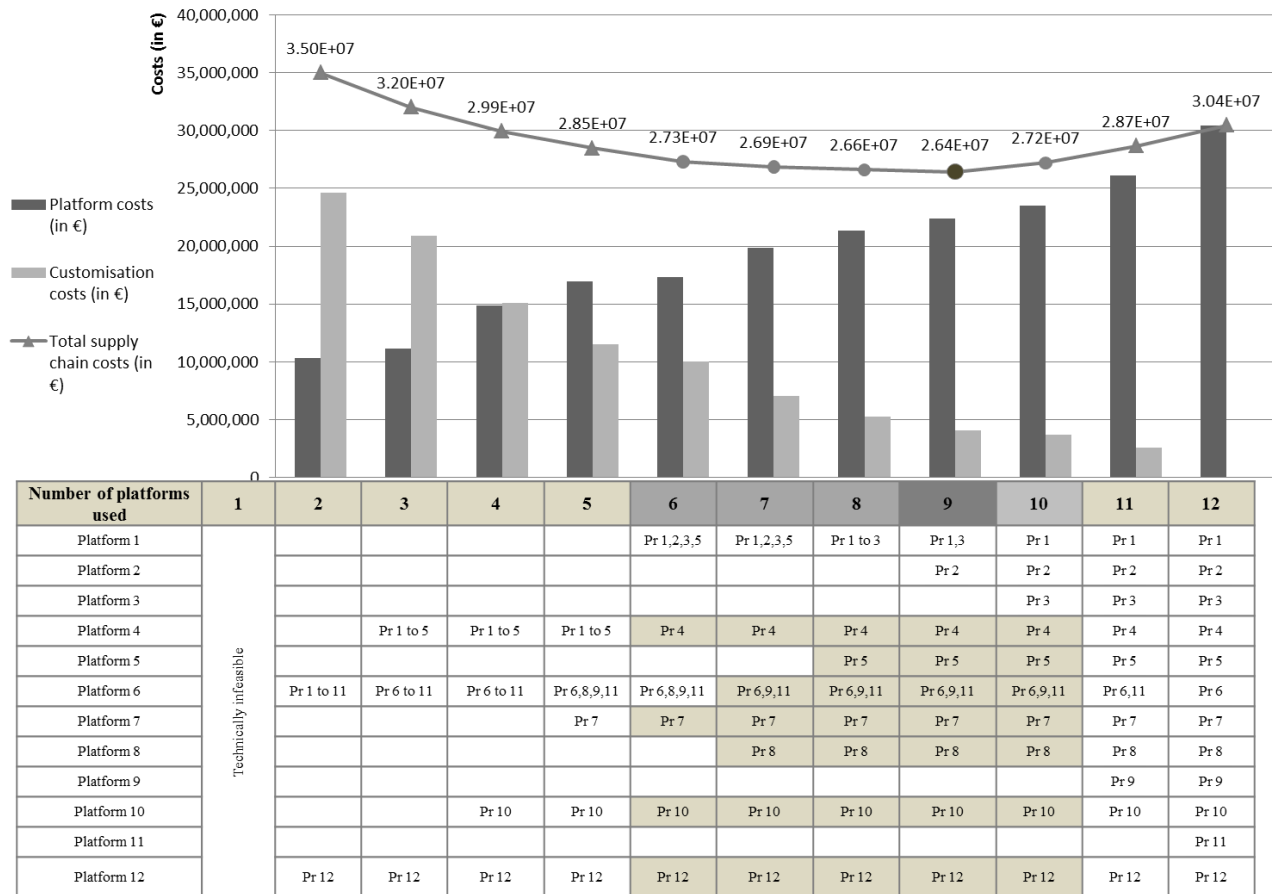


Figure 4: Optimal product-platform decisions given the number of platforms used

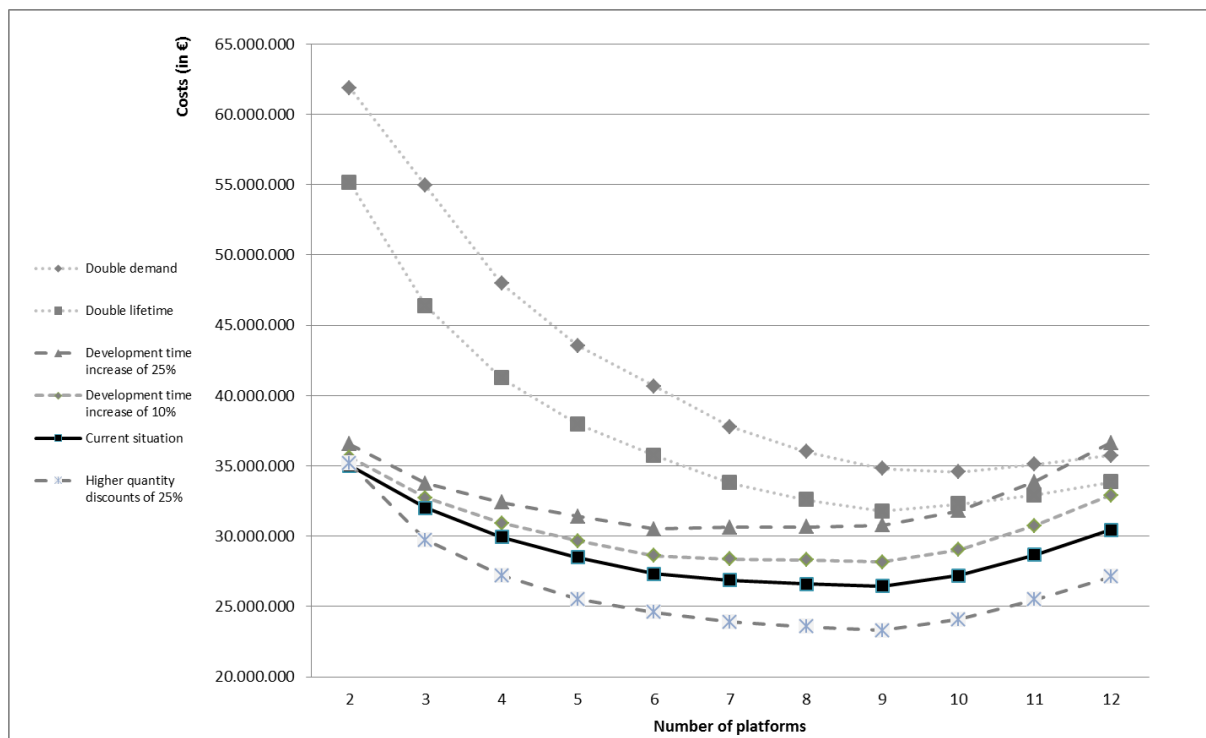


Figure 5: Impact of input parameter changes